LET'S BUILD TOMORROW TODAY
Cisco Nexus 7000 / 7700 Switch Architecture

Tim Stevenson
Distinguished Engineer, Technical Marketing
BRKARC-3470
Session Abstract

This session presents an in-depth study of the architecture of the latest generation of Nexus 7000 and Nexus 7700 data center switches. Topics include supervisors, fabrics, I/O modules, forwarding engines, and physical design elements, as well as a discussion of key hardware-enabled features that combine to implement high-performance data center network services.
Session Goal

• To provide a thorough understanding of the Nexus 7000 / Nexus 7700 switching architecture, chassis, supervisor, fabric, and I/O module design, key forwarding engine functions, and complete packet flows

• This session will examine the Nexus 7700 system, as well as the latest additions to the Nexus 7000

• This session will not examine NX-OS software architecture or other Nexus platform architectures
Agenda

- Introduction to Nexus 7000 / Nexus 7700
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Fabric Architecture
- Hardware Forwarding
- Packet Walks
- Conclusion
Introduction to Nexus 7000 / Nexus 7700 Platform

Data-center class Ethernet switches designed to deliver high performance, high availability, system scale, and investment protection

Designed for wide range of Data Center deployments, focused on feature-rich 10G/40G/100G density and performance
Nexus 7000 / Nexus 7700 – Common Foundation

**Nexus 7000**
General purpose DC switching w/10/40/100G

**Nexus 7700**
Targeted at Dense 40G/100G deployments

Common Foundation
• Same release vehicles, versioning, feature-sets
• Common configuration model
• Common operational model

NX-OS
• Common fabric ASICs (Fab2) and architecture
• Same central arbitration model
• Same VOQ/QOS model

Fabrics
• Identical forwarding ASICs (F2E, F3)
• Consistent hardware feature sets
• Consistent hardware scale
Agenda

• Introduction to Nexus 7000 / Nexus 7700
• Chassis Architecture
• Supervisor Engine and I/O Module Architecture
• Fabric Architecture
• Hardware Forwarding
• Packet Walks
• Conclusion
Nexus 7000 Chassis Family

Nexus 7010

Nexus 7018

Nexus 7009

Nexus 7004

NX-OS 4.1(2) and later

NX-OS 6.1(2) and later

NX-OS 5.2(1) and later
Nexus 7700 Chassis Family

Nexus 7718
- 26RU
- N77-C7718
- NX-OS 6.2(2) and later

Nexus 7710
- 14RU
- N77-C7710
- NX-OS 6.2(2) and later

Nexus 7706
- 9RU
- N77-C7706
- NX-OS 6.2(6) and later
Nexus 7702 Chassis

Front:
- One Supervisor Engine
- Two Power Supplies
- One F3 Series I/O Module

Rear:
- One Fan Tray (3 Fans)
- No Fabric Modules!

NX-OS 7.2 and later

3RU
Agenda

- Introduction to Nexus 7000 / Nexus 7700
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Fabric Architecture
- Hardware Forwarding
- Packet Walks
- Conclusion
Supervisor Engine 2 / 2E

- Provides all control plane and management functions

<table>
<thead>
<tr>
<th>Supervisor Engine 2 (Nexus 7000)</th>
<th>Supervisor Engine 2E (Nexus 7000 / Nexus 7700)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base performance</td>
<td>High performance</td>
</tr>
<tr>
<td>One quad-core 2.1GHz CPU with 12GB DRAM</td>
<td>Two quad-core 2.1GHz CPU with 32GB DRAM</td>
</tr>
</tbody>
</table>

- Connects to fabric via 1G inband interface
- Interfaces with I/O modules via 1G switched EOBC
- Onboard central arbiter ASIC
  Controls access to fabric bandwidth via dedicated arbitration path to I/O modules
Supervisor Engine 2 / 2E Architecture
Reference: Component Functions – Supervisor Engines

- Main CPU(s) – Runs all system-level NX-OS processes and handles all control plane and management functions

- Switched EOBC – Provides switch 1G connections to each module CPU for internal system management and communication

- I/O Controller – Provides all I/O functions for supervisor components

- VOQs – Interface to central arbiter and local crossbar fabric, implements Virtual Output Queuing

- Fabric ASIC – Local fabric that provides first/third stage of three-stage crossbar

- Central Arbiter – Dedicated ASIC that controls access to fabric based on destination interface and priority of requests
Nexus 7000 / 7700 I/O Module Families

M Series Modules

- M1 1G and 10G
- M2 10G / 40G / 100G

F Series Modules

- F1 10G
- F2 10G
- F2E 10G
- F3 10G / 40G / 100G

L2/L3/L4 with large forwarding tables and rich feature set

High performance, low latency with streamlined feature set

F3 closes the F/M feature gap!
Nexus 7000 M2 I/O Modules
N7K-M224XP-23L / N7K-M206FQ-23L / N7K-M202CF-22L

- 10G / 40G / 100G M2 I/O modules
- Share common hardware architecture – multi-chipset
- Two integrated forwarding engines (120Mpps)
- Layer 2/Layer 3 forwarding with L3/L4 services (ACL/QOS) and advanced features (MPLS/OTV/GRE etc.)
- Large forwarding tables (900K FIB/128K ACL)
- 802.1AE LinkSec on all ports

<table>
<thead>
<tr>
<th>Module</th>
<th>Port Density</th>
<th>Optics</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2 10G</td>
<td>24 x 10G (plus Nexus 2000 FEX support)</td>
<td>SFP+</td>
<td>240G</td>
</tr>
<tr>
<td>M2 40G</td>
<td>6 x 40G (or up to 24 x 10G via breakout)</td>
<td>QSFP+</td>
<td>240G</td>
</tr>
<tr>
<td>M2 100G</td>
<td>2 x 100G</td>
<td>CFP</td>
<td>200G</td>
</tr>
</tbody>
</table>
Nexus 7000 M2 I/O Module Architecture
N7K-M224XP-23L / N7K-M206FQ-23L / N7K-M202CF-22L
Reference: ASIC Functions – M2 Modules

- LinkSec + MAC – Provides port ASIC functions, including buffering/queuing, and performs 802.1ae encryption/decryption for front-panel ports
- Replication Engine – Bridge between front panel port, forwarding engine, and fabric; performs multicast and SPAN replication
- Forwarding Engine – Performs all Layer 2, Layer 3, and Layer 4 forwarding decisions and policy enforcement
- VOQs – Interface to central arbiter and local crossbar fabric, implements Virtual Output Queuing
- Arbitration Aggregator – Muxes arbitration requests from VOQs before sending to central arbiter on Supervisor Engine
- Fabric ASIC – Local fabric that provides first/third stage of three-stage crossbar
- (LC CPU – Linecard CPU, runs module-specific NX-OS processes and interfaces with Supervisor Engine over EOBC)
M2 Module 40G and 100G Flow Limits

• M2 modules use 10G Virtual Queuing Index (VQI)
• Each VQI sustains 10G traffic flow
• All packets in given 5-tuple flow hash to single VQI using port-channel load-balancing algorithm
• Single-flow limit is 10G
Nexus 7000 / Nexus 7700 F2E I/O Modules
N7K-F248XP-25E / N7K-F248XT-25E / N77-F248XP-23E

- 48-port 1G/10G with SFP/SFP+ transceivers
- 48-port 1G/10GBaseT with RJ-45 connectors (Nexus 7000)
- 480G full-duplex fabric connectivity
- System-on-chip (SOC) forwarding engine design
  - 12 independent SOC ASICs
- Layer 2/Layer 3 forwarding with L3/L4 services (ACL/QoS)
- Interoperability with M1/M2, in Layer 2 mode on Nexus 7000
  - Proxy routing for inter-VLAN/L3 traffic
Nexus 7000 F2E Module Architecture

N7K-F248XP-25E / N7K-F248XT-25E

Front Panel Ports (SFP/SFP+): 4 X 10G SOC 1 to 4 X 10G SOC 12

To Fabric Modules:
- EOBC
- LC CPU
- LC Inband
- to ARB
- to ARB

To Central Arbiters:
- Arbitration Aggregator
- to LC CPU

Fabric ASIC
Nexus 7700 F2E Module Architecture
N77-F248XP-23E
Reference: ASIC Functions – F2E Modules

• 4 X 10G SoC – Four-port 10G system-on-chip; provides Port ASIC, Replication Engine, Forwarding Engine, and VOQ functions

• Arbitration Aggregator – Muxes arbitration requests from SoCs before sending to central arbiter on Supervisor Engine

• Fabric ASIC – Local fabric that provides first/third stage of three-stage crossbar

• (LC CPU – Linecard CPU, runs module-specific NX-OS processes and interfaces with Supervisor Engine over EOBC)
Nexus 7000 F3 I/O Modules
N7K-F348XP-25 / N7K-F312FQ-25 / N7K-F306CK-25

- 10G / 40G / 100G F3 I/O modules
- Share common hardware architecture
- SOC-based forwarding engine design
  - 6 independent SOC ASICs per module
- Layer 2/Layer 3 forwarding with L3/L4 services (ACL/QOS) and advanced features (FP/MPLS/OTV/GRE/VXLAN etc.)
- Require Supervisor Engine 2 / 2E

<table>
<thead>
<tr>
<th>Module</th>
<th>Port Density</th>
<th>Optics</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3 10G</td>
<td>48 x 1/10G (plus Nexus 2000 FEX support)</td>
<td>SFP+</td>
<td>480G</td>
</tr>
<tr>
<td>F3 40G</td>
<td>12 x 40G (or up to 48 x 10G via breakout)</td>
<td>QSFP+</td>
<td>480G</td>
</tr>
<tr>
<td>F3 100G</td>
<td>6 x 100G</td>
<td>CPAK</td>
<td>550G</td>
</tr>
</tbody>
</table>
Nexus 7700 F3 I/O Modules
N7K-F348XP-25 / N7K-F312FQ-25 / N7K-F306CK-25

- 10G / 40G / 100G F3 I/O modules
- Share common hardware architecture
- SOC-based forwarding engine design
  - 6 independent SOC ASICs per 10G module
  - 12 independent SOC ASICs per 40G/100G module
- Layer 2/Layer 3 forwarding with L3/L4 services (ACL/QOS) and advanced features (FP/MPLS/OTV/GRE/VXLAN etc.)

<table>
<thead>
<tr>
<th>Module</th>
<th>Port Density</th>
<th>Optics</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3 10G</td>
<td>48 x 1/10G (plus Nexus 2000 FEX support)</td>
<td>SFP+</td>
<td>480G</td>
</tr>
<tr>
<td>F3 40G</td>
<td>24 x 40G (or up to 76 x 10G + 5 x 40G via breakout)</td>
<td>QSFP+</td>
<td>960G</td>
</tr>
<tr>
<td>F3 100G</td>
<td>12 x 100G</td>
<td>CPAK</td>
<td>1.2T</td>
</tr>
</tbody>
</table>
Nexus 7000 F3 48-Port 1G/10G Module Architecture
N7K-F348XP-25
Fabric Services Accelerator (FSA) for F3

- High-performance module CPU with on-board acceleration engines
  - 6Gbps inband connectivity from SOCs to FSA
  - Multi-Mpps packet processing
  - 2 X 2GB dedicated DRAM

- Performance/scale boost for distributed fabric services, including sampled Netflow and BFD (roadmap)

- Other potential applications include distributed ARP/ping processing, data plane packet analysis (wireshark), network probing, etc.
Nexus 7000 F3 12-Port 40G Module Architecture
N7K-F312FQ-25

Front Panel Ports (QSFP+)

- 2 X 40G SOC 1
- 2 X 40G SOC 2
- 2 X 40G SOC 3
- 2 X 40G SOC 4
- 2 X 40G SOC 5
- 2 X 40G SOC 6

Fabric ASIC

- 1G switch
- EOBC
- To Fabric Modules

Arbitration Aggregator
- To Central Arbiters
- x 6

EOBC Inband to ARB x 6
Nexus 7000 F3 6-Port 100G Module Architecture
N7K-F306CK-25
Nexus 7700 F3 48-Port 1G/10G Module Architecture

N77-F348XP-23
Nexus 7700 F3 24-Port 40G Module Architecture
N77-F324FQ-25

Front Panel Ports (QSFP+)

Fabric ASIC

To Fabric Modules

Fabric ASIC

To Central Arbiters

Arbitration Aggregator

EOBC

FSA CPU

1G switch

LC Inband

to ARB

2 X 40G SOC 1
2 X 40G SOC 2
2 X 40G SOC 3
2 X 40G SOC 4
2 X 40G SOC 5
2 X 40G SOC 6
2 X 40G SOC 7
2 X 40G SOC 8
2 X 40G SOC 9
2 X 40G SOC 10
2 X 40G SOC 11
2 X 40G SOC 12

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Nexus 7700 F3 12-Port 100G Module Architecture
N77-F312CK-26
Reference: ASIC Functions – F3 Modules

• 8 X 10G / 2 X 40G / 1 X 100G SOC – 10/40/100G capable system-on-chip; provides Port ASIC, Replication Engine, Forwarding Engine, and VOQ functions

• Arbitration Aggregator – Muxes arbitration requests from SoCs before sending to central arbiter on Supervisor Engine

• Fabric ASIC – Local fabric that provides first/third stage of three-stage crossbar

• (FSA CPU – Fabric Services Accelerator, a linecard CPU with built-in application acceleration for higher performance BFD, sampled Netflow, and other functions; runs module-specific NX-OS processes and interfaces with Supervisor Engine over EOBC)
F3 Module 40G and 100G Flows

- Virtual Queuing Index (VQI) sustains 10G, 40G, or 100G traffic flow based on destination interface type
- No single-flow limit – full 40G/100G flow support
I/O Module Interoperability

• General module interoperability rule is: “+/−1 generation” in same Virtual Device Context (VDC)

• Two main module interoperability models:
  • “Proxy Forwarding”
  • “Ingress Forwarding” with Lowest Common Denominator
Proxy Forwarding Model

**M2 + F2E VDC**

- F2E modules run in pure Layer 2 mode – all L3 functions disabled
- M2 modules host SVIs and other L3 functions
- From F2E perspective, Router MAC reachable via M2 modules
- All packets destined to Router MAC forwarded through fabric toward one M2 module, selected via port-channel hash
- M2 modules(s) perform all L3 forwarding and policy, pass packets back over fabric to output port
- **Key consideration:** M-series L3 routing capacity versus F-series front-panel port count – How much Layer 3 routing is required?

```
interface vlan 10
  ip address 10.1.10.1/24

interface vlan 20
  ip address 10.1.20.1/24
```

---

```
Host A
  10.1.10.100
  vlan 10
  
Host B
  10.1.20.100
  vlan 20
```

---

```Mac Table
rtr-mac → M2 modules```

**Port-channel hash selects M2 module**

**Router MAC reached via M2 modules**
Ingress Forwarding with Lowest Common Denominator Model

F3 + M2 VDC -or- F3 + F2E VDC

• F3 module interoperability always “Ingress Forwarding” – NO proxy forwarding
  • Ingress module receiving packet makes all forwarding decisions for that packet

• Supported feature set and scale based on Lowest Common Denominator
  • Feature available if all modules support the feature
  • Table sizes based on lowest capacity

<table>
<thead>
<tr>
<th>Module Types in VDC</th>
<th>Layer 2</th>
<th>Layer 3</th>
<th>VPC</th>
<th>MPLS</th>
<th>OTV</th>
<th>Fabric Path</th>
<th>VXLAN</th>
<th>Table Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F3 size</td>
</tr>
<tr>
<td>F3 + M2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>F3 size</td>
</tr>
<tr>
<td>F3 + F2E</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>F2E size</td>
</tr>
<tr>
<td>M2 + F2E + F3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>Not supported</td>
</tr>
</tbody>
</table>
## Interoperability Forwarding Model Matrix

<table>
<thead>
<tr>
<th>Module Combination</th>
<th>Interoperability Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 + M2</td>
<td>Lowest Common Denominator</td>
</tr>
<tr>
<td>M + F1</td>
<td>Proxy Forwarding</td>
</tr>
<tr>
<td>M + F2E</td>
<td>Proxy Forwarding</td>
</tr>
<tr>
<td>F2 + F2E</td>
<td>Lowest Common Denominator</td>
</tr>
<tr>
<td>F2 + F2E + F3</td>
<td>Lowest Common Denominator</td>
</tr>
<tr>
<td>M2 + F3</td>
<td>Lowest Common Denominator</td>
</tr>
<tr>
<td>M2 + F2/F2E + F3</td>
<td>NOT SUPPORTED</td>
</tr>
</tbody>
</table>
Agenda

• Introduction to Nexus 7000 / Nexus 7700
• Chassis Architecture
• Supervisor Engine and I/O Module Architecture
• Fabric Architecture
• Hardware Forwarding
• Packet Walks
• Conclusion
Crossbar Switch Fabric Modules

- Provide interconnection of I/O modules
- Nexus 7000 and Nexus 7700 fabrics based on Fabric 2 ASIC
- Each installed fabric increases available per-payload slot bandwidth

<table>
<thead>
<tr>
<th>Fabric Module</th>
<th>Supported Chassis</th>
<th>Per-fabric module bandwidth</th>
<th>Max fabric modules</th>
<th>Total bandwidth per slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nexus 7000 Fabric 2</td>
<td>7009 / 7010 / 7018</td>
<td>110Gbps per slot</td>
<td>5</td>
<td>550Gbps per slot</td>
</tr>
<tr>
<td>Nexus 7700 Fabric 2</td>
<td>7706 / 7710 / 7718</td>
<td>220Gbps per slot</td>
<td>6</td>
<td>1.32Tbps per slot</td>
</tr>
</tbody>
</table>

- Different I/O modules leverage different amount of available fabric bandwidth
- Access to fabric bandwidth controlled using QOS-aware central arbitration with VOQ
Multistage Crossbar

Nexus 7000 / Nexus 7700 implement 3-stage crossbar switch fabric

- Stages 1 and 3 on I/O modules
- Stage 2 on fabric modules
I/O Module Capacity – Nexus 7000

550Gbps
per slot bandwidth

One fabric:
• **Any port** can pass traffic to **any other port** in VDC

Three fabrics:
• 240G M2 module has maximum bandwidth

Five fabrics:
• 480G F2E/F3 module has maximum bandwidth
What About Nexus 7004?

- Nexus 7004 has no fabric modules
- Each I/O module has local fabric with 10 available fabric channels
  - I/O modules connect “back-to-back” via 8 fabric channels
  - Two fabric channels “borrowed” to connect supervisor engines
I/O Module Capacity – Nexus 7700

1320Gbps
per slot bandwidth

One fabric:
- **Any port** can pass traffic to **any other port** in VDC

Three fabrics:
- 480G F2E/F3 10G module has maximum bandwidth

Five fabrics:
- 960G F3 40G module has maximum bandwidth

Six fabrics:
- 1.2T F3 100G module has maximum bandwidth
What About Nexus 7702?

- Nexus 7702 has no fabric modules
- Single I/O module – all traffic locally switched
- Two fabric channels connect to supervisor engine
Agenda

- Introduction to Nexus 7000 / Nexus 7700
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Fabric Architecture
- Hardware Forwarding
- Packet Walks
- Conclusion
Hardware Forwarding Process

- Ingress queuing and scheduling
- Perform forwarding lookups
- Forward through fabric
- Egress queuing and scheduling
Ingress / Egress Queuing and Scheduling

- Nexus 7000 / Nexus 7700 use queuing policies and network-QOS policies to define queuing and scheduling behavior

- Default policies always in effect in absence of any user configuration

- Queuing model varies by I/O module family
  - **M-series modules**: hybrid model combining egress-buffered architecture with ingress-buffered architecture
  - **F-series modules**: pure ingress-buffered architecture
M2 – Hybrid Ingress/Egress Buffered

**Ingress Buffering**

- **8 ingress queues per port**
- **Ingress port buffer** – Buffers traffic for congested forwarding/replication engines and congested egress destinations (VQIs)
- **Virtual queuing** – Congestion management and local scheduling toward egress destinations (VQIs)
- **Independent scheduling for each destination/priority**

**Multistage Crossbar Fabric**

- VOQ ASIC
  - VOQ buffer carved by source and priority
  - Ingress buffer – Buffers traffic for congested egress destinations (VQIs)
- Egress Buffer
  - Egress buffer – Receives frames from fabric and schedules traffic toward egress ports
  - 4 priority levels per VQI

**Egress Buffering**

- **Egress port buffer** – Manages congestion at egress ports
- **8 egress queues per port**

**FABRIC-QOS POLICY**

- Ingress Queuing Policies
- Egress Queuing Policies
F2E/F3 – Ingress Buffered

Ingress Buffering

- Independent scheduling for each destination/priority
- Virtual queuing – Congestion management and local scheduling toward egress destinations (VQIs)

Multistage Crossbar Fabric

- Central Arbiter

Ingress Buffering

- Virtual Queuing
- Ingress Buffer – Buffers traffic for congested egress destinations (VQIs)

Egress Buffering

- Egress buffer – Receives frames from fabric and schedules traffic toward egress ports
- 4/8 priority levels per port (VQI)

Network-QoS Policy

- Ingress Queuing Policy
- Egress Queuing Policy
F3 Buffer Sharing

Default – Dedicated per Port Buffer

- All ingress buffer carved equally among ports
- One port cannot use another port’s unused buffer

Optional – Shared Buffer + Dedicated per Port Buffer

- Total buffer split into dedicated per-port portion and shared portion
- Any port can consume buffer in shared portion
- Increases burst absorption, assuming not all ports burst simultaneously

CiscoLive!

10G F3 used as example
Hardware Forwarding Lookups

• Layer 2 and Layer 3 packet flow virtually identical in hardware

• Forwarding engine / decision engine pipeline provides consistent L2 and L3 lookup performance

• Pipelined architecture also performs ingress and egress ACL, QOS, and Netflow lookups, affecting final forwarding result
M2 Forwarding Engine Hardware

- Two hardware forwarding engines integrated on every M2 I/O module
- Layer 2 switching (with hardware MAC learning)
- Layer 3 IPv4/IPv6 unicast and multicast
- MPLS/VPLS/EoMPLS

<table>
<thead>
<tr>
<th>Hardware Table</th>
<th>M-Series Modules without Scale License</th>
<th>M-Series Modules with Scale License</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address Table</td>
<td>128K</td>
<td>128K</td>
</tr>
<tr>
<td>Classification TCAM (ACL/QOS)</td>
<td>64K</td>
<td>128K</td>
</tr>
<tr>
<td>Netflow Table</td>
<td>1M</td>
<td>1M</td>
</tr>
</tbody>
</table>
F2E Forwarding Engine Hardware

- 4 x 10G SOC with decision engine
- Layer 2 switching (with hardware MAC learning)
- Layer 3 IPv4/IPv6 unicast and multicast
- FabricPath forwarding
- RACL/VACL/PACL
- QOS remarking and policing policies
- Ingress sampled Netflow

<table>
<thead>
<tr>
<th>Hardware Table</th>
<th>F2E Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address Table</td>
<td>16K</td>
</tr>
<tr>
<td>FIB TCAM</td>
<td>32K IPv4/16K IPv6</td>
</tr>
<tr>
<td>Classification TCAM (ACL/QOS)</td>
<td>16K</td>
</tr>
</tbody>
</table>
F3 Forwarding Engine Hardware

- 8 x 10G, 2 x 40G, or 1 x 100G SOC with decision engine
- Layer 2 switching (with hardware MAC learning)
- Layer 3 IPv4/IPv6 unicast and multicast
- FabricPath forwarding

- RACL/VACL/PACL
- QOS remarking and policing policies
- Ingress sampled Netflow
- MPLS/VPLS/EoMPLS
- OTV / GRE tunnels
- LISP
- VXLAN

<table>
<thead>
<tr>
<th>Hardware Table</th>
<th>F3 Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Address Table</td>
<td>64K</td>
</tr>
<tr>
<td>FIB TCAM</td>
<td>64K IPv4/32K IPv6</td>
</tr>
<tr>
<td>Classification TCAM (ACL/QOS)</td>
<td>16K</td>
</tr>
</tbody>
</table>
F2E/F3 Decision Engine

Ingress lookup pipeline:
- Receive packet from Port Logic block
- Send payload to Ingress Buffer
- Send header to Decision Engine

Classification (ACL/QOS/SNF):
- Ingress ACL/QOS/SNF classification

Layer 3 FIB:
- FIB TCAM and adjacency table lookups for Layer 3 forwarding
- ECMP hashing

Policing:
- Ingress policing
- Egress policing

Final Results:
- Return final result (destination + priority) to Ingress Buffer

L2 Lookup (pre-L3):
- Ingress MAC table lookups

L2 Lookup (post-L3):
- Egress MAC lookups

Egress lookup pipeline:
- Egress MAC table lookups
- Port-channel hash result

Ingress Parser:
- From Ingress Port Logic
- Send payload to Ingress Buffer
- Send header to Decision Engine

F2E/F3 SOC
Layer 2 Hardware Forwarding

- Layer 2 forwarding driven by MAC table lookups
- Source and destination MAC lookups performed for each frame, based on \{VLAN,MAC\} pairs
- Source MAC lookup drives new learns and refreshes aging timers
- Destination MAC lookup returns destination port
MAC Table Lookup

From Ingress Parser

(VLAN,MAC) pair fed into two separate hash functions

Extract VLAN and MAC address

Ingress frame header

MAC Table Hash #1

Ingress frame

MAC Table Hash #2

Bank 1

Bank 2

MAC Table (2*16*2048 = 64K entries)

16 pages

Hash result #1 selects row on each page of Bank 1

Hash result #2 selects row on each page of Bank 2

Matched entry returns destination index

Compare (VLAN,MAC) pair to selected row on each page of each bank

To Ingress Lookup Pipeline

F3 module used as example
MAC Table Details

• MAC table is a hash table
• Effective capacity of any hash table dependent on “quality” of input to hash
• Hash collisions can occur and 100% utilization may not be possible
• Hash collisions more likely when many MACs are similar, e.g.:
  Performance testers (all ports sending incrementing MAC addresses)
  Virtualized environments (new VMs deployed en masse with incrementing MACs)
  Homogenous server environments (all hosts using the same NICs from the same vendor and all purchased at roughly the same time)
  IP multicast deployments (many L2 group MACs which start with 01005E and likely to have sequential destination IP group addresses)
Layer 3 Hardware Forwarding

- Layer 3 forwarding driven by FIB table lookups
- Forwarding tables built by control plane on supervisor engine
  - OSPF, EIGRP, IS-IS, BGP, statics, etc.
- Tables downloaded to forwarding engine hardware for data-plane forwarding
- FIB TCAM lookup based on longest-match destination IP prefix lookup
- FIB match returns rewrite (next-hop) information in adjacency table
IP FIB TCAM Lookup

Generate lookup key based on destination IP and compare to FIB TCAM entries

Flow data from packet header fed into load-sharing hash function

Modulo of hash result and # next-hops selects exact ADJ entry

Return lookup result
Classification Lookups

• Matching packets
  • Layer 2, Layer 3, and/or Layer 4 information

• Used to decide whether to apply a particular policy to a packet
  • Enforce security, QOS, or other policies

• Some examples:
  • Match TCP/UDP source/destination port numbers to enforce security policy
  • Match source IP addresses to apply policy-based routing (PBR)
  • Match 5-tuple to apply QOS marking policy
  • Match protocol-type to apply Control Plane Policing (CoPP)
  • etc.
CL TCAM Lookup – ACL

Packet header

From Ingress/Egress Lookup Pipeline

If hit in CL TCAM returns result in CL SRAM

Classification

Actions from security ACL

Decision Engine

Security ACL

ip access-list example
permit ip any host 10.1.2.100
deny ip any host 10.1.68.44
deny ip any host 10.33.2.25
deny ip any host 10.24.77.7
deny ip any host 10.24.77.1
permit tcp any any eq 22
deny tcp any any eq 23
deny udp any any eq 514
permit tcp any any eq 80
permit udp any any eq 161

ACEs from security ACL (x = don’t care)

Packet header header

Generate lookup key based on packet fields and compare to CL TCAM entries

Mask out “don’t care” bits while comparing key

Hit in CL TCAM returns result in CL SRAM

Fields to match: src IP | dst IP | protocol | src port | dst port

Hit!

Return lookup result, affecting final packet handling (forward or drop)

To Ingress/ Egress Lookup Pipeline

HIT!

Decision Engine

CL TCAM

CL SRAM

Generate lookup key based on packet fields and compare to CL TCAM entries

 perk ip any host 10.1.2.100
deny ip any host 10.1.68.44
deny ip any host 10.33.2.25
deny ip any host 10.24.77.7
deny ip any host 10.24.77.1
permit tcp any any eq 22
deny tcp any any eq 23
deny udp any any eq 514
permit tcp any any eq 80
permit udp any any eq 161

HIT!
CL TCAM Lookup – QOS

Generate lookup key based on packet fields and compare to CL TCAM entries

Packet header

From Ingress/Egress Lookup Pipeline

CL TCAM

Classification

Hit in CL TCAM returns result in CL SRAM

Remark DSCP 32

Remark DSCP 32

Remark DSCP 40

Remark DSCP 40

Remark IP Prec 3

Remark IP Prec 3

Hit!

ACEs from QOS policy ACLs

(x = don’t care)

Decision Engine

Actions from QOS policy

Return lookup result, affecting final packet handling (police or remark)

Return lookup result, affecting final packet handling (police or remark)

QOS Policy ACLs

ip access-list police
permit ip any 10.3.3.0/24
permit ip any 10.4.12.0/24
permit ip any 10.10.1.0/24
permit ip any 10.20.2.0/24
ip access-list remark-dscp-32
permit udp 10.0.1.0/24 any
permit udp 10.1.1.0/24 any
ip access-list remark-dscp-40
permit tcp 10.0.1.0/24 any
permit tcp 10.1.1.0/24 any
ip access-list remark-prec-3
permit tcp any 10.2.3.0/24 eq 23
permit tcp any 10.5.5.0/24 eq 23

QOS Policy ACLs

ip access-list police
permit ip any 10.3.3.0/24
permit ip any 10.4.12.0/24
permit ip any 10.10.1.0/24
permit ip any 10.20.2.0/24
ip access-list remark-dscp-32
permit udp 10.0.1.0/24 any
permit udp 10.1.1.0/24 any
ip access-list remark-dscp-40
permit tcp 10.0.1.0/24 any
permit tcp 10.1.1.0/24 any
ip access-list remark-prec-3
permit tcp any 10.2.3.0/24 eq 23
permit tcp any 10.5.5.0/24 eq 23
Full and Sampled Netflow

Netflow collects flow data for export to collector(s)

**Full** Netflow: Accounts for every packet of every flow on interface, up to capacity of hardware Netflow table

- Available on M-Series modules only

**Sampled** Netflow: Accounts for M in N packets on interface using random packet-based sampling

- **M2**: Accounts sampled flows, up to capacity of hardware Netflow table
- **F2E/F3**: Accounts hardware sampled flows in software
M2 Netflow Table

Netflow “Table” actually consists of three hardware components in M2 forwarding engine:

- **Netflow Hash Table**: Contains Netflow Entry Keys and corresponding indexes to Netflow Entry Table (speeds lookups and minimizes hash collisions)
- **Netflow Entry Table**: Contains actual Netflow flow data
- **Netflow Statistics Table**: Contains statistics for corresponding flow entries
NDE on M2 Modules

LC CPU builds NDE packets (IP+UDP+NDE) and sends them to Sup via EOBC.

Sup CPU transmits NDE packets either via mgmt0 or via Inband to collector(s).

Supervisor Engine

Fabric Modules

EOBC

LC CPU periodically ages out Netflow table entries.

Flow entries created/updated in Netflow table (full or sampled).

Hardware Flow Creation

Data-plane traffic traverses forwarding engines on each module.

Forwarding Engine

L3 Engine

L2 Engine

Replication Engine

Port ASIC

Fabric

VOQs

M2 Module

Sup CPU

via mgmt0

via Supervisor Inband

Supervisor Inband

Fabric

NF Table

L3 Engine

L2 Engine

Replication Engine

Port ASIC
F2E/F3 Sampled Netflow

- Hardware-based sampling with software-based Netflow cache
- Classification lookup selects Netflow sampler-table entry
- Sampler table defines which sampler to use (defines M:N)
- Copy of randomly sampled packets sent to LC CPU/FSA via module inband
  - Sampled copies “sliced” to reduce bandwidth consumption
- Sampled copies rate-limited to protect LC CPU
  - F2E to ~1000pps per module
  - F3 to ~3000pps per module
  - Roadmap for ~50Kpps per F3 module using FSA
LC CPU builds software Netflow cache based on samples and periodically ages out entries.

Packets marked for sampling copied to LC inband.

Samples subjected to HW rate limiter.

Sampler marks M:N random packets to sample.

Classification block in Decision Engine selects sampler.
Agenda

- Introduction to Nexus 7000 / Nexus 7700
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Fabric Architecture
- Hardware Forwarding
- Packet Walks
- Conclusion
M2 System Architecture

Supervisor Engine

Central Arbiter

Fabric Module 1
Fabric Module 2
Fabric Module 3
Fabric Module 4
Fabric Module 5

Fabric ASIC

Virtual Queuing

Ingress Buffer

Forwarding Engine 1

FIB
CL
TCAM
NF
Table

L3 Engine

MAC
Table

L2 Engine

Module 1

VOQ 1

VOQ 2

VOQ 3

VOQ 4

FE 1

FE 2

FE 3

FE 4

Internal Ports

Port ASIC 1

Port ASIC 2

Port ASIC 3

Port ASIC 4

Port ASIC 5

Local Ports

10G M2 module used as example
M2 Packet Flow
F2E/F3 Packet Flow

Virtual Queuing
- Queue packet descriptor in VOQ (destination port + priority)
- Payload queued in ingress buffer based on COS / DSCP

Ingress Buffer
- Receive packet from wire
- CRC, storm control, VLAN translation, etc.
- Ingress packet parsing
- If PKT and HDR
- Packet headers sent to DE
- Layer 2 Lookups
- Layer 3 Lookups
- Classification for ACL / QOS / SNF

Decision Engine
- MAC Table
- FIB
- TCAM
- CL TCAM
- Final lookup result: destination port + priority

Supervisor Engine
- Central Arbiter
- Buffer credit granted
- Credit
- N7700 10G F3 module used as example

Port Logic
- Request buffer credit for destination port + priority
- Transmit to fabric

Egress Buffer
- Receive from fabric
- Buffer on egress based on destination port + priority
- Schedule for transmission
- Transmit on wire
- VLAN translation, etc.

Module 1
- Fabric Module 1
- Fabric Module 2
- Fabric Module 3
- Fabric Module 4
- Fabric Module 5
- Fabric Module 6
- Fabric ASIC

F2E/F3 Packet Flow
Agenda

• Introduction to Nexus 7000 / Nexus 7700
• Chassis Architecture
• Supervisor Engine and I/O Module Architecture
• Fabric Architecture
• Hardware Forwarding
• Packet Walks
• Conclusion
Conclusion

• You should now have a thorough understanding of the Nexus 7000 / Nexus 7700 switching architecture – including chassis, supervisors, I/O modules, and fabrics, as well as forwarding lookups and complete system-level packet flows…

• Any questions?
Reference: Acronym Decoder

- ACL—Access Control List
- ADJ—Adjacency
- ASIC—Application Specific Integrated Circuit
- CFP—C Formfactor Pluggable
- CoPP—Control Plane Policing
- COS—Class of Service
- DE—Decision Engine
- DSCP—Differentiated Services Code Point
- DWRR—Deficit Weighted Round Robin
- ECMP—Equal Cost Multi Path
- EOBC—Ethernet Out-of-Band Channel
- FCoE—Fiber Channel over Ethernet
- FE—Forwarding Engine
- FEX—Fabric Extender (Nexus 2000 family)
- FIB—Forwarding Information Base
- GRE—Generic Route Encapsulation
- MPLS—Multicontext Switching
- NDE—Netflow Data Export
- OTV—Overlay Transport Virtualization
- PACL—Port ACL
- PBR—Policy-Based Routing
- QOS—Quality of Service
- QSFP+—40G Quad Small Formfactor Pluggable
- RAAL—Router ACL
- RE—Replication Engine
- RPF—Reverse Path Forwarding
- RU—Rack Unit
- SFP+—10G Small Formfactor Pluggable
- SNF—Sampled Netflow
- SOC—System-on-chip/switch-on-chip
- SP—Strict priority (queue)
- TCAM—Ternary CAM
- VACL—VLAN ACL
- VDC—Virtual Device Context
- VOQ—Virtual Output Queuing
- VQI—Virtual Queuing Index
- VXLAN—Virtual Extensible LAN
Participate in the “My Favorite Speaker” Contest
Promote Your Favorite Speaker and You Could Be a Winner

• Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)

• Send a tweet and include
  • Your favorite speaker’s Twitter handle
  • Two hashtags: #CLUS #MyFavoriteSpeaker

• You can submit an entry for more than one of your “favorite” speakers

• Don’t forget to follow @CiscoLive and @CiscoPress

• View the official rules at http://bit.ly/CLUSwin
Complete Your Online Session Evaluation

• Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.

• Complete your session surveys though the Cisco Live mobile app or your computer on Cisco Live Connect.

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings
- Related sessions
Thank you
TOMORROW starts here.